Text normalization

Why I talk about this...

Written and spoken-domain language

Many speech and language technologies require a mapping between “written” and “spoken” representations of language.

- In *text normalization* we map pseudo-ideographic representations like $4.20 to pronounceable representations like *four dollars and twenty cents*.
- In *transliteration* we map from one orthographic code like スパゲッティ to another like *supagetti*.
- In *grapheme-to-phoneme conversion* (or “g2p”) we map from an orthographic code (e.g., *tuviéramos*) to a phonemic one like [tubieramos].

Text normalization

- Currency expressions: $4.20 → *four dollars and twenty cents*
- Date expressions: 11/2 → *November second*
- Letter sequences: WinNT → *win n_letter t_letter*
- Numbers: 69 → *sixty nine*
- Measure expressions: 12kg → *twelve kilograms*

Following Taylor (2009), we refer to these categories as *semiotic classes*, and their conversion as *text normalization*.

Text normalization systems require some degree of linguistic sophistication.

A taxonomy of semiotic classes (Sproat et al. 2001)

	EXPN	abbreviation	*adv, N.Y, mph, gov't*
alpha	LSEQ	letter sequence	*CIA, D.C, CDs*
	ASWD	read as word	*CAT*, proper names
	MSPL	misspelling	*geogaphy*
	NUM	number (cardinal)	*12, 45, 1/2, 0·6*
	NORD	number (ordinal)	*May 7, 3rd, Bill Gates III*
	NTEL	telephone (or part of)	*212 555-4523*
	NDIG	number as digits	*Room 101*
N	NIDE	identifier	*747, 386, I5, pc110, 3A*
U	NADDR	number as street address	*5000 Pennsylvania, 4523 Forbes*
M	NZIP	zip code or PO Box	*91020*
B	NTIME	a (compound) time	*3·20, 11:45*
E	NDATE	a (compound) date	*2/2/99, 14/03/87* (or US) *03/14/87*
R	NYER	year(s)	*1998, 80s, 1900s, 2003*
S	MONEY	money (US or other)	*$3·45, HK$300, Y20,000, $200K*
	BMONEY	money tr/m/billions	*$3·45 billion*
	PRCT	percentage	*75%, 3·4%*
	SPLT	mixed or "split"	*WS99, x220, 2-car* (see also SLNT and PUNC examples)
	SLNT	not spoken,	word boundary or emphasis character:
M		word boundary	*M.bath, KENT*RLTY, _really_*
I	PUNC	not spoken,	non-standard punctuation: "***" in
S		phrase boundary	*$99,9K***Whites*, "…" in *DECIDE…Year*
C	FNSP	funny spelling	*sllooooooww, sh*t*
	URL	url, pathname or email	*http://apj.co.uk, /usr/local, phj@tpt.com*
	NONE	should be ignored	ascii art, formatting junk

Taxonomy of semiotic classes (Ebden & Sproat 2015)

- Cardinal: 69 → *sixty nine*
- Date: 11/2/1985 → *November second nineteen eighty five*
- Decimal: 23.3 → *twenty three point three*
- Electronic: kgorman@gc.cuny.edu → *k gorman at gc dot cuny dot edu*
- Fraction: 2/5 → *two fifths*
- Measure: 12kg → *twelve kilograms*
- Money: $5.96 → *five dollars and ninety six cents*
- Ordinal: 69th → *sixty ninth*
- Roman numeral: LIV → *fifty four*
- Telephone: 215-566-6123 → *two one five, five six six, six one two three*
- Time: 11:58 → *eleven fifty eight*

Taxonomy of semiotic classes (van Esch & Sproat 2017)

Table 2: *Word-like tokens.*

EXPN — abbreviation	adv, N.Y, mph, govt
LSEQ — letter sequence	CIA, D.C., CDs
MSPL — misspelling	geogaphy
FNSP — funny spelling	sllooooooww, cul8r
CENSORED	sh*t, f***
ASWD — verbalized as word	CAT, NATO, LASER, GOOG
1337-speak	1337, n00b
SPLT	C-SPAN
AMPERSAND-WORD	AT&T, H&M
MUSIC-NOTE	C, D, Re
TRANSLIT	words in another script

Table 3: *Basic numbers.*

NUM — cardinal numbers	12, 45, -5, Superbowl XLVIII
NORD — ordinal numbers	May 7, 3rd, Bill Gates III
NDIG — number as digits	Room 101,
DECIMAL	1.34, -1.34

Taxonomy of semiotic classes (van Esch & Sproat 2017)

Table 4: *Identifiers.*

NIDE — identifier	747, 386, I5, pc110, 3A U, A4, 401(k)'s
NTEL — telephone	212 555-4523
Version numbers	Android 3.1, 5.2.13, 5.3.alpha
Years in sports club names	Astana 1964
Credit card numbers	1243 4567 8910 1234
PIN numbers, CCV	1234
Social Security Numbers	123-45-6789
ID numbers	S123456789
Seasons/episodes	S02E02
Flight numbers	AAL12, Flight 637
Call signs	808BL (*eight oh eight bravo lima*)
Runways	13L / 13R (*thirteen left*, *thirteen right*)
URL — url, pathname or email	http://apj.co.uk, /usr/local, phj@tpt.com, foobar.jpg, test1.js
MENTION	#hashtag, @person, +person

Table 5: *Dates and times.*

NTIME — a (compound) time	3:20 p.m., 11:45, 15:38, 3 o'clock, 3pm, 2:02:57, 8ish
NDATE — a (compound) date	2/2/99, 14/03/87 (or US) 03/14/87, March 3rd 2019, Mar 3, 6/87, 31.VII.1932
NYER — year(s)	1998, 80s, 1900s, 2003, '70s, John Smith '19, 29 BCE,
Durations	3hr30m, 4h2m23s, 2:02:57, 24/7/365
Time zones	GMT, UTC+1, UTC+5:45

Taxonomy of semiotic classes (van Esch & Sproat 2017)

Table 6: *Ratios, percentages and fractions.*

PRCT — percentage	75%, 3.4%, -3%
Fractions	$\frac{1}{3}$, 1/3 , 1 $\frac{1}{2}$, 1 1/2
Ratios	2:3, 3.37:1
Ratings	4.5/5, **** (*four stars*)

Table 7: *Geographic entities.*

NADDR building numbers	5000 Pennsylvania, 4523 Forbes
NZIP zip code	91020, 23945-2345, 1039 AA, 2
PO Box	PO Box 1
Lat-long	52°22′N 4°54′E
Street numbers	W 17th St, Hougang Street 21, 14 St, 8 Ave
Within-building numbers	#301, apt 301
Area/grid numbers	1200 South, Pier 39
Exit numbers	Exit 314 towards AZ/W141
Provinces/states/countries	OH, CA, MA, QLD, NB, USA, UK, MX, BR
Highway numbers	I-280, S101, A113, Route 66, CA 17, US 101, I35W

Taxonomy of semiotic classes (van Esch & Sproat 2017)

Table 8: *Measure expressions.*

Measures	km, 3 mi, ft-lbs
Square measures	km^2, 10 acres, 10 ha
Cubic measures	30 cu ft, 30 m^3
Relative measures	km/h, mph, 4,034/km^2, 234 mpg, 10 l/100km
Meas. with punct.	6”, 5’8”
Temperatures	0 C, 5°, -5°
Dimensions	1024x768, 10x4x8
Stock indices	. . . opened 17,652.36 (insert “points”)
Wire gauge	1/0 (*one aught*), 2/0 (*two aught*)

Table 9: *Sports-related expressions.*

Plain scores	3-1 *three to one*
Tennis	15-0 (*fifteen love*)
Australian football	10.12 (72) (*ten twelve seventy two*)
Chess notation	Nc6, Rxc6

Table 10: *Currency expressions.*

MONEY — money	$3.45, HK$300, Y20,000
BMONEY — money tr/m/billions	$3.45 billion

Wikipedia (“written” domain)

The giraffe has an extremely elongated neck, which can be up to 2 m (6 ft 7 in) in length, accounting for much of the animal's vertical height. Each cervical vertebra is over 28 cm (11 in) long. They comprise 52–54 percent of the length of the giraffe's vertebral column, compared with the 27–33 percent typical of similar large ungulates, including the giraffe’s closest living relative, the okapi.

Wikipedia (“spoken” domain)

The giraffe has an extremely elongated neck, which can be up to two meters (six feet seven inches) in length, accounting for much of the animal's vertical height. Each cervical vertebra is over twenty eight centimeters (eleven inches) long. They comprise fifty two to fifty four percent of the length of the giraffe's vertebral column, compared with the twenty seven to thirty three percent typical of similar large ungulates, including the giraffe’s closest living relative, the okapi.

Wikipedia (“written” domain)

The giraffe has an extremely elongated neck, which can be up to **2 m (6 ft 7 in)** in length, accounting for much of the animal's vertical height. Each cervical vertebra is over 28 cm (11 in) long. They comprise 52–54 percent of the length of the giraffe's vertebral column, compared with the 27–33 percent typical of similar large ungulates, including the giraffe’s closest living relative, the okapi.

Wikipedia (“spoken” domain)

The giraffe has an extremely elongated neck, which can be up to **two meters** (**six feet seven inches**) in length, accounting for much of the animal's vertical height. Each cervical vertebra is over twenty eight centimeters (eleven inches) long. They comprise fifty two to fifty four percent of the length of the giraffe's vertebral column, compared with the twenty seven to thirty three percent typical of similar large ungulates, including the giraffe’s closest living relative, the okapi.

Applications of text normalization

- In text-to-speech synthesis (TTS), the front-end is responsible for providing pronunciations for semiotic classes and out-of-vocabulary (OOV) words
  - This can then be fed into the back-end (e.g., WaveNet, or a parametric synthesizer)
- In automatic speech recognition (ASR):
  - The text used to train the language model are converted to spoken form
  - Spoken form outputs from the recognizer are converted back to written form so they can be displayed to the user
- In information extraction (IE), verbalizations can be used as a canonical form for spoken and the various written forms of dates, times, etc.

Prior work on “noisy text”

There is quite a bit of prior work focusing on novel abbreviations (e.g., `cn u plz hlp?`) such as found in informal genres (e.g., microblogs, SMS).

But if you want to prepare text for downstream processing, there is little need to verbalize number or measure phrases, but this is *essential* for speech applications.

We are interested in the semiotic categories most relevant to speech, and as such are using *text normalization* in the original sense of the term (Sproat et al. 2001).

Text normalization is hard

Not all errors are equal:

- E.g., it may be acceptable to read plz → *plaza* when the context actually requires it be read as *please*…
- But is definitely not acceptable to read 72 → *four hundred seventy two*

We refer to the latter kind of error as *silly errors*.

And some errors cannot really be handled without world knowledge: e.g., Dr. Baltimore, MD → *doctor baltimore maryland* (?)

Machine learning for text normalization at Google

- Sentence boundary detection (Sproat & Hall 2014)
- English abbreviation expansion (Roark & Sproat 2014, Gorman et al. 2021)
- Grapheme-to-phoneme prediction (Jansche 2014, Rao et al. 2015, van Esch et al. 2016)
- Russian word stress prediction (Hall & Sproat 2013)
- Letter sequence prediction (Sproat & Hall 2014)
- Homograph disambiguation (Gorman, Mazovetskiy & Nikolaev 2018, Seale 2021)
- End-to-end research (Ng, Gorman, & Sproat, et al. 2017, Sproat & Jaitly 2017, Zhang et al. 2019)

But...

Nearly all text normalization is done with hand-written language-specific grammars (just as it was over 20 years ago: e.g., Sproat 1996).

Grammar-based systems are particularly interpretable, but require linguistic sophistication and substantial development efforts to achieve acceptable performance.

We need to support languages for which finding a literate, linguistically-sophisticated native speaker-cum-software engineer is nigh impossible.

For speech tech, this is **the** major barrier to quality control and internationalization.

Minimally supervised number normalization

Gorman, Kyle and Sproat, Richard. 2016. Minimally supervised number normalization. *Transactions of the Association for Computational Linguistics* 4: 507-519.

Why numbers?

Nearly all semiotic classes contain numbers.

Therefore, when adding a new language, one of the first things we write is a number grammar.

At the start of this project (late 2015), we had hand-written number grammars for roughly 70 languages/locales.

These can be difficult to write: they require a linguistically sophisticated native speaker (or a sophisticated linguist with high-quality reference materials).

Why this ought to work

It *seems like*, if I tell you that in French, 7 is *sept*, and 97 is *quatre vingt dix sept*, you ought to know how to say 90, too.

Then, if I tell you that 8 is *huit*, you ought to know how to say 98, and so on.

Something ought to be able to learn to do this.

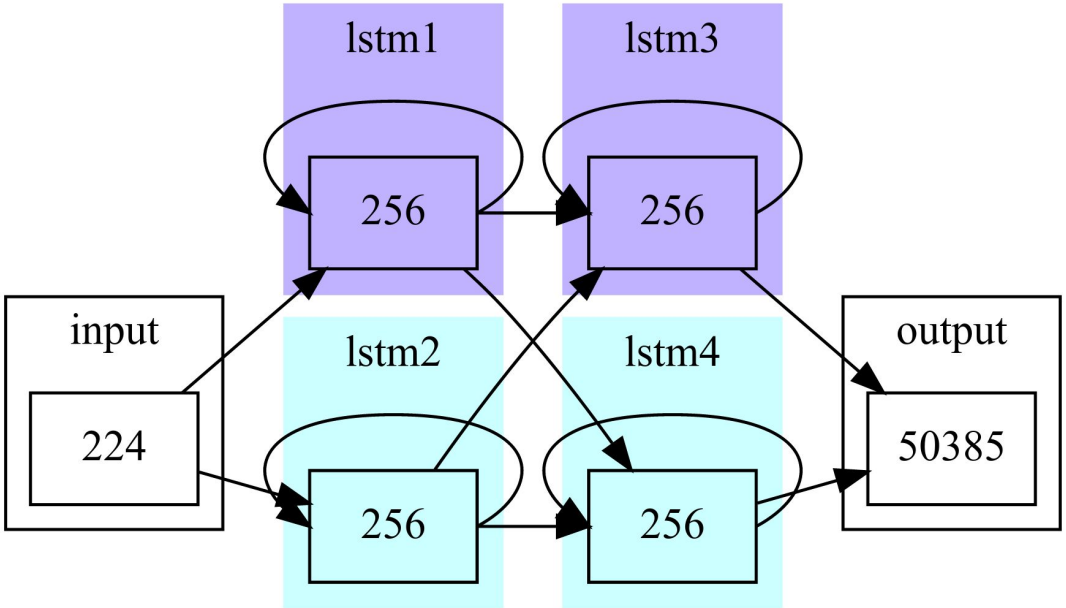
Similarly, the developmental literature on number cognition focuses on the emergence of a *successor* function which gives you the $(i + 1)$th number.

Local outline

- Number normalization with recurrent neural networks and large amounts of training data
- Number normalization with finite-state transducers and very small amounts of training data

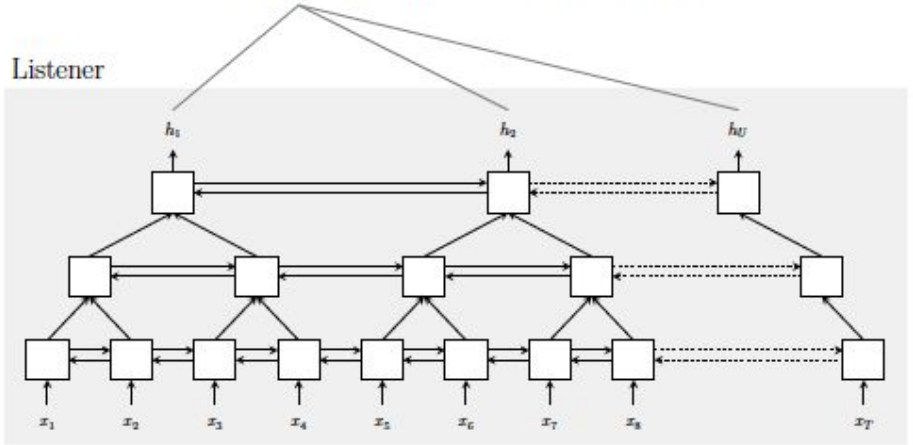
Noisy channel LSTM

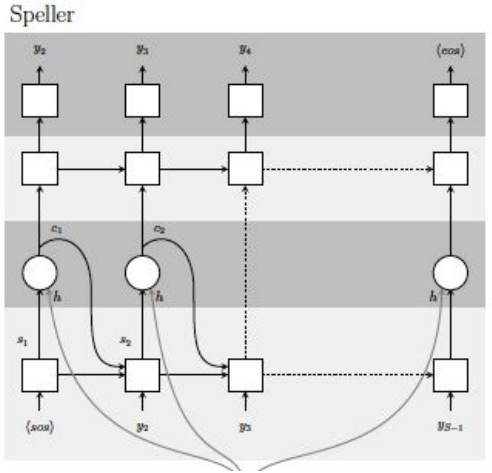
A channel model consisting of four hidden feed-forward LSTM layers (two forwards, two backwards) and a CTC output layer with softmax activation (cf. Rao et al. 2015)



Attention-based RNN

Four layer pyramidal encoder with 256 attentional units and a two-layer decoder (Chan et al. 2016)





Data set

Large: 28,000 cardinal numbers extracted from several terabytes of Russian web text using our production TTS front-end (Ebden & Sproat 2015)

Medium: 10,000 numbers randomly sampled (without replacement) from a modified Yule-Simon distribution (10% held out for evaluation)

Small: 300 "curated" examples, consisting of [0, 200] and 99 large numbers

(NB: We will reuse the **Medium** and **Small** in later evaluations.)

Results

Training size	LSTM accuracy	Attention accuracy	Overlap
28,000	.999	1.000	56%
9,000	.994	1.000	0%
300	< .001	< .001	< 1%

Error analysis

With the **Medium** and **Large** data sets, *all* errors are of the silly variety:

E.g., 9801**:**

девять	тысяч	семьсот	один	(Hypothesis)
nine	thousand	seven.hundred	one	
девять	тысяч	восемьсот	один	(Gold)
nine	thousand	eight.hundred	one	

With the **Small** data set, nothing of use is learned.

Hand-written number grammars

The traditional approach involves the composition of two components:

1. A language-specific *factorization* FST *F*:

   97000 → 4 20 10 7 1000

2. A language-specific *verbalization* FST *L*:

   4 20 10 7 1000 → *quatre vingt dix sept mil*

Verbalization FST *L*

Ignoring morphological concord, building *L* is trivial:

…
4 → *quatre*
7 → *sept*
10 → *dix*
20 → *vingt*
1e3 → *mil*
…

….
60 → шестьдесят
60 → шестьдесяти
60 → шестьдесятью
…

For languages like Russian, we use a finite-state language model to disambiguate.

Factorization FST *F*

This isn’t so clear: what's the hypothesis space?

Numeral bases (*WALS* #131)

Decimal	125
Hybrid vigesimal-decimal	22
Pure vigesimal	20
Other base	5
Extended body-part system	4
Restricted	20

http://wals.info/chapter/131

(Brazilian) Portuguese: *decimal*

sete	*milhões*	*quinhentos*	*e*	*quarenta*	*e*	*três*	*mil*
seven	million	five.hundred	and	forty	and	three	thousand

= 7 x 1,000,000 + 500 + 40 + 3 x 1000 = 7,543,000

trezentos	*e*	*vinte*	*e*	*um*
three.hundred	and	twenty	and	one

= 300 + 20 + 1 = 321

Georgian: *hybrid vigesimal-decimal*

ots	*da*	*tsamet'i*
twenty	and	thirteen

= 20 + 13 = 33

ormots	da	tekvsmet'i
forty	and	sixteen

= 40 + 16 = 56

otkh-	*asi*
four	hundred

= 4 x 100 = 400

Other hybrid vigesimal-decimal

French:

quatre-vingt-dix-sept
four-twenty-ten-seven
= 4 x 20 + 10 + 7 = 97

Danish:*

halv-tred-sind-s-tyve
half-third-times-of-twenty
= (3 - ½) x 20 = 50

* This is an archaic/formal form; *halvtreds* is the informal one.

San Mateo Huave (Stairs Kreger & Scharfe de Stairs 1985:398-399): *pure vigesimal*

nimiow	*gajpowüw*
twenty	ten

= 20 + 10 = 30

ic	*miow*
two	twenty

= 2 x 20 = 40

piquiuw	*acoic*	*miow*
four	five	twenty

= 4 x 5 x 20 = 400

Ekari (Drabbe 1952:30): *other base*

èna	*ma*	*gàati*	*dàimita*	*mutò*
one	and	ten	and	sixty

= 1 + 10 + 60 = 71

Assamese: "South Asian" powers of ten

es	100
ek haajaa	1,000
dah haajaa	10,000
ek laakh	100,000
dah laakh	1,000,000
ek kooti	10,000,000
dah kooti	100,000,000

Addend flop (Sproat 2000:191)

German:

zwei	*hundert*	*acht-und-neunzig*
two	hundred	eight-and-ninety

= 2 x 100 + 8 + 90 = 298

Malagasy:

efatra	*amby*	*valopolo*	*sy*	*eninjato*	*sy*	*telo*	*arivo*
four	rest	eighty	and	six.hundred	and	three	thousand

= 4 + 80 + 6 x 100 + 3 x 1000 = 3,684

Kinyarwanda: multiplicand flop

mangana	*abiri*	*na*	*mirongo*	*itatu*
hundred	two	and	ten	three

= 100 x 2 + 10 x 3 = 230

igihumbi	*cumi*	*na*	*bibiri*
thousand	ten	and	two

= 1,000 x 10 + 2 = 12,000

Mandarin: creative use of zero

萬	零	五	十
10,000	0	5	10

= 10,000 + 0 + 5 x 10 = 10,050

Rare number operations

Latin: backcounting:

un-de-viginti
one-from-twenty
= 20 - 1 = 19

Welsh: multiplication by a half:

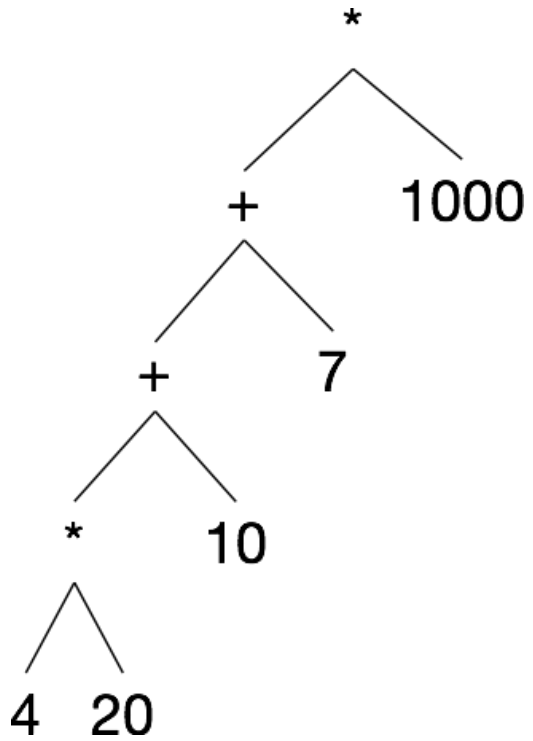
hanner cant a phedwar
half hundred and four
= ½ x 100 + 4 = 54

A universal grammar for number names (Hurford 1975)

- Additions of two or more integral addends...
- ...multiplication of two or integral addends...
- ...and occasional "flops", subtractions, etc.
- ...and morphological concord.

The expressions themselves are CFG-equivalent.

We just need to learn the rules and compile them.



Hurford, James. 1975. *The linguistic theory of numerals*. Cambridge: CUP.

Induction

Induction is made easier by the fact that we know the numeric denotation and the relation is constrained by basic arithmetic.

Given training data of the form:

80	*quatre vingt*
97	*quatre vingt dix sept*

We induce a grammar *G* as follows.

Derivation of *quatre-vingt-dix-sept*

97

quatre *vingt* *dix* *sept*

Derivation of *quatre-vingt-dix-sept*

97

90 + 7

80 + 10 + 7

4 x 20 + 10 + 7

...

quatre *vingt* *dix* *sept*

Derivation of *quatre-vingt-dix-sept*

97

90 + 7

80 + 10 + 7

4 x 20 + 10 + 7

...

4 — *quatre*

20 — *vingt*

10 — *dix*

7 — *sept*

Derivation of *quatre-vingt-dix-sept*

97

90 + 7 | 80 + 10 + 7 | 4 x 20 + 10 + 7 | ...

4 + 20 + 10 + 7 | 4 x 20 + 10 + 7 | 4 + 20 x 10 + 7 | ...

4 | 20 | 10 | 7

quatre | *vingt* | *dix* | *sept*

Derivation of *quatre-vingt-dix-sept*

97

90 + 7 | 80 + 10 + 7 | **4 x 20 + 10 + 7** | ...

4 + 20 + 10 + 7 | **4 x 20 + 10 + 7** | 4 + 20 x 10 + 7 | ...

4 | 20 | 10 | 7

quatre | *vingt* | *dix* | *sept*

Rule extraction

We extract syntactic rules from this intersection, which usually contains just one analysis, to create a grammar *G*:

S → (7 | 10 | 4 | 20 | * | +)
* → 4 20
+ → * 10 7

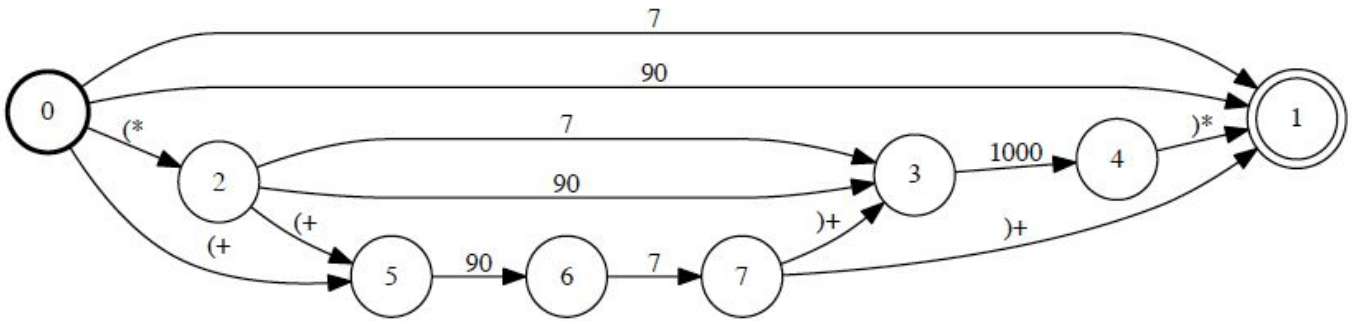
Some ambiguities remain...

Putting it all together...

The grammar is realized as *G*, a pushdown transducer. Then our final model is:

$$F \circ M \circ G \circ L$$

F: language-universal factorization transducer
M: language-universal markup deletion transducer



Results

Locale	Training size	Accuracy	Overlap
en_us	9,000	1.000	0%
	300	1.000	< 1%
ka_ge	9,000	1.000	0%
	300	1.000	< 1%
km_kh	9,000	1.000	0%
	300	1.000	< 1%
ru_ru	28,000	1.000	56%
	9,000	0.998	0%
	300	0.998	< 1%

Error analysis

Fortunately, *all* remaining errors are of the non-silly variety.

E.g., 70477170:

семьдесят	миллион	…	(Hypothesis)
seven.hundred	million.nom.sg.	…	
семьдесят	миллионов	…	(Gold)
seven.hundred	million.gen.pl.	…	

Roll-out

A Google linguist (supported by native annotators, program managers, and code reviewers) can create roughly one number grammar per day.

We have created number grammars for more than 100 languages using the method.

We continue squash bugs and expand our covering grammars.

Local summary

Two models for number normalization:

- Recurrent neural networks with minimal domain knowledge but substantial supervision
- A cascade of induced finite-state transducers with considerable domain knowledge but minimal supervision

Subsequent work (Ritchie et al. 2019)

- Generalizations of the covering grammars
- UniNum: an open-source number-name database with entries for 186 languages, locales, and scripts.

Improving homograph disambiguation with machine learning

Gorman, Kyle, Mazovetskiy, Gleb, and Nikolaev, Gleb.2018. Improving homograph disambiguation with machine learning. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, pages 1349-1352.

What is a homograph?

Homographs are words that are pronounced differently depending on the intended sense, e.g.:

> *read* [ɹiːd] (present tense) and *read* [ɹɛd] (past tense).
> *lead* [liːd] (show the way) and *lead* [lɛd] (metal).

- Not merely polysemous: *bank* has multiple senses but only one pronunciation.
- Not just pronunciation variation: *the* can be realized as [ðə] or [ðiː] depending on context and degree of emphasis, but there's no semantic distinction.

Importance for TTS

Humans are extremely sensitive to errors in homographs disambiguation.

Homographs are very common in certain languages. E.g., in English, ≈5% of the words in Google TTS requests (as of 2017) are homographs.

Homographs are also a very common source of bugs in the TTS system itself: #2 in English, #1 in Russian.

Homograph taxonomy

Morphosyntactic homographs: morphosyntactic distinctions not indicated in the orthography; e.g., *abuse*, *read*.

Lexical homographs: accidental overlap in the spelling of semantically unrelated lexemes; e.g., *bow*, *sake*.

Mixed homographs: morphosyntactic distinctions accompanied by semantic distinctions; e.g., *consummate*, *produce*.

This cuts across a traditional distinction between those homographs which can be disambiguated on the basis of part of speech and those which cannot.

Rule-based homograph disambiguation

1. *wind* followed by *up* or *down* is `wind_vrb` [waɪnd].
2. *wind* preceded by *strong* is `wind_nou` [wɪnd].

Rule types

- Context (words before / after)
- Substring / regex
- Morphosyntax / POS / proper noun
- Geolocation match
- Default

Only the first matching rule is applied.

Pros and cons

Pros:

- Rules are expressive and can describe anything you can compute
- Rules are interpretable
- Rules support simple "point fixes" easy

Cons:

- Rules are expensive to write and maintain
- Writing rules for lexical homographs can be difficult (e.g., *bow*)
- Many edge cases

Example

winds:

`winds_nou`: [wɪndz]
`winds_vrb`: [waɪndz]

Rules:

- If the following word is *up* or *down*, resolve to `winds_vrb`.
- Default: `winds_nou`.

"Today, there will be winds up to 30 mph."

Our approach

Much prior work (Hearst 1991; Sproat et al. 1992; Yarowsky 1997, Silva et al. 2012):

- Identify candidate contextual cues
- Compute log-odds, log-likelihoods, etc. for these cues
- Build a decision list
- Prune (via cross-validation, subsumption, etc.)

Our approach:

- Identify candidate contextual cues
- Discriminatively train regularized classifiers with these cues as features

Baseline features: context words

Context unigrams, bigrams, and skipgrams. E.g.:

"In 1995 I **read** Lord of the Rings."

Φ_w = { WL2:<DATE>, WL1:i, WR1:lord, WR2:of,
WL2:<DATE>_WL1:i, WR1:lord_WR2:of, WL1:i_WR1:lord }

Note that we use equivalence classes for context semiotic classes tokens.

Baseline features: target POS tag

POS tag on the target homograph. E.g.:

"In 1995 I **read** Lord of the Rings."

Φ_t = { WT=VBN }

The POS tagger does not currently run on embedded engines (those that run on mobile devices which cannot quickly connect to the WAN) so this feature is inactive there.

Baseline features: target capitalization

Unicode capitalization category for the target homograph (Sproat et al. 1992).

"In 1995 I **read** Lord of the Rings."

Φ_c = { C=lower }

Inference

Extracts features then assigns the most probable resolution.

At runtime, we just need to compute, for each word ID *i*:

$$\text{argmax}_x \, w_x \cdot \phi = \text{argmax}_x \sum_i w_{x,i} \, \phi_i$$

Training

Multinomial* log-linear ("maxent") model, used for other classification and ranking tasks in the TTS front-end (Hall & Sproat 2013, Sproat & Hall 2014).

We use gradient descent with the FTRL optimizer, L_1 regularization, and Vizier (Golovin et al. 2017) for hyperparameter tuning.

We train separate models for each homograph.

*Why multinomial? There may be more than two word IDs per homograph.

Model hybridization

Two ways to incorporate ML:

- Delete all rules and just use the ML model.
- Allow existing rules to pre-empt the ML model (i.e., treat the ML prediction as the default rule).

Initial data collection

We identified a set of roughly 200 (now whittled down to 163) `en_us` homographs for targeted annotation.

Then, we randomly sampled 200 sentences per homograph from English Wikipedia which contained them, then manually filtered these down to 100 per homograph.

Annotation crowd-sourced internally to native speakers, 3 annotations per homographs, conflicting annotations resolved by linguists.

This high-quality dataset of ≈16,000 examples is freely available.

Baseline evaluation

	Micro-accuracy	Macro-accuracy
MAP baseline	.850	.849
Embedded: rules	.869	.863
Server: rules	.893	.890

Held-out evaluation (90%/10%)

	Micro-accuracy	Macro-accuracy
Embedded: rules	.870	.867
Server: rules	.890	.886
Embedded: ML	.926	.924
Server: ML	.954	.951
Embedded: rules + ML	**.990**	**.990**
Server: rules + ML	**.990**	**.990**

Error analysis

Morphosyntactic homographs like *read* are more challenging than lexical homographs like *bass*, and this primarily accounts for the server/embedded disparity.

Still many POS tag errors in sentences like "Smith has played Trophy matches for the county from 1993 to *present*." These cause some downstream errors.

Subsequent work (Seale 2021)

- Manual error correction of the Gorman et al. data
- Experiments with various "muppet" transformer models
- Projecting weakly-labeled training data using MT alignments:

  E.g., in Russian *bass* is translated as either *бас* 'bass (music part or instrument)' and *окунь* 'perch', depending on sense.

- Projecting weakly-labeled training data using ASR alignments:

  E.g., if the recognizer says that the word was pronounced as [bæs], it is probably the fish sense, not the music sense.

Structured abbreviation expansion in context

Gorman, Kyle, Kirov, Christo, Roark, Brian, and Sproat, Richard. 2021. Structured abbreviation expansion in context. In *Findings of the Association for Computational Linguistics: EMNLP 2021,* pages 995-1005.

Motivations

Text normalization, transformations to prepare text for downstream processing (particularly speech recognition or synthesis), requires the handling of *abbreviations*, among other semiotic classes.

High-frequency, highly-conventionalized abbreviations (e.g., *mL*, *lbs*, *AK*, *NZ*) are often expanded using hand-written grammars, possibly augmented with machine learning for contextual disambiguation.

We are instead interested in infrequent, non-conventionalized *ad hoc abbreviations*, common on communication channels which favor brevity.

Trajan's column (113 CE)
[Image credit: Britannica]

[Image credit: Wikipedia]

SENATVS POPVLVSQVE ROMANVS
IMP CAESARI DIVI NERVAE F NERVAE
TRAIANO AVG GERM DACICO PONTIF
MAXIMO TRIB POT XVII IMP VI COS VI P P
AD DECLARANDVM QVANTAE ALTITVDINIS
MONS ET LOCVS TAN[tis oper]IBVS SIT EGESTVS

Prior work

Abbreviation expansion has been studied using data from SMS, chatrooms, and social media platforms such as Twitter.

Most of these studies use small, manually curated data sets in which the "ground truth" is produced by annotators asked to expand abbreviations using local context.

> However, Baldwin et al. (2015) report that this annotation task produces poor inter-annotator agreement.

And, unfortunately, nearly all of the publicly-released data sets are no longer available, perhaps for licensing reasons.

Data collection

We produce an unencumbered data set using a task in which annotators *generate* (rather than disambiguate) abbreviated text.

- Sentences are extracted from English-language Wikipedia pages.
- A character LM is used to rank the sentences on per-character entropy, and sentences from the middle of the distribution are sampled for annotation.
- Annotators are asked to shorten the sentences by at least 20 characters.
- Exploratory analysis of annotators' abbreviation expansion strategies broadly accords with our intuitions about English.
- A separate team of annotators were able to disambiguate the abbreviated text with a high degree of accuracy.

Task definition

Let **A** = [a_0, a_1, …, a_n] be an *n*-length sequence of possibly-abbreviated words.

Let **E** = [e_0, e_1, …, e_n] be an *n*-length sequence of expanded words.

If e_i is an element of **E**, then it is either:

> not an abbreviation: a_i is identical to e_i, or
> an abbreviation: a_i is a proper (non-null) subsequence of e_i.

This limits us to what Pennell & Liu (2010) call *deletion-based abbreviation*.

This yields a highly-tractable task definition, which can be relaxed in future work.

Hmm, getting a strong "noisy channel" vibe here...

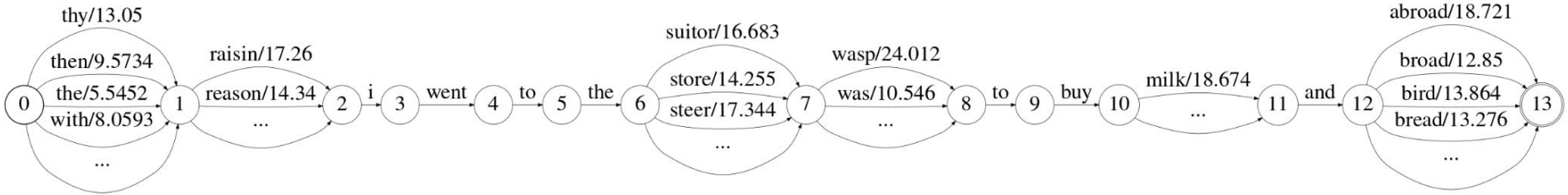
Finite-state pipeline

The finite-state pipeline is defined by two weighted finite state automata:

- $P(\mathbf{E})$: a *language model*, defining a probability distribution over expansions
- $P(a_i, e_i)$: a *pair n-gram model*, defining the joint probability over abbreviation/expansion string pairs, estimated with an online, batched *Viterbi training* variant of expectation maximization.

These models are fused to assemble a lattice of candidate expansions and decoded via the shortest path algorithm.

th rsn i went to the str ws to buy mlk and brd .



Neural pipeline

The neural pipeline substitutes an LSTM language model for the expansion model. We also added some heuristics to the generation of expansion candidates:

- **LexBlock:** If a_i is in-vocabulary, set the probability of non-faithful expansion candidates to zero.
- **Memory:** do not prune candidate e_i if it occurs as an expansion of a_i in the training set.
- **SubBlock:** if candidate e_i is a contiguous substring of another candidate e_i', set the probability of the superstring candidate to zero.

These models are fused and decoded via left-to-right approximate beam search.

Held-out evaluation (80%/10%/10%)

	WER	OER	UER	IER
n-gram LM, pair LM	2.90	0.00	2.13	4.08
LSTM LM, pair LM	1.41	0.39	0.19	2.35
LSTM LM, subseq.	**1.12**	0.40	0.20	1.74
Human topline	3.51	2.23	0.30	4.88

Error analysis

- Roughly 40% of errors are "harmful":
  - the **{clases, ✓ classes, ✗ clashes}** cntinud nd th band strugld fr time to rite tgthr
  - anothr criticism is abt th absenc o a stndrd **{auditin, ✓ auditing, ✗ audition}** procedr .
- American vs. British spellings (both are present in Wikipedia):
  - consequently th village hs develpd a mor suburbn role than som o its **{neighbrs, ✓ neighbours, ✗ neighbors}**
- Morphologically related expansions:
  - they **{recog, ✓ recognized, ✗ recognize}** accomps by musicians frm th prev yr .
- Syntactically-similar expansions:
  - **{th, ✓ the, ✗ this}** behavr s strengthnd by an automatc reinfrcng consequenc .

Future work

- Relax the restriction to deletion-based abbreviation
- Experiment with more powerful language models
- Survey abbreviation strategies in languages beyond English

End-to-end research

Generalized text normalization with covering grammars

Ng, Gorman & Sproat (2017) perform text normalization on English and Russian using finite-state covering grammars (including the number grammars induced with the method just described) and a log-linear ranker to select the best verbalization.

Sproat & Jaitly (2017) and Zhang et al. (2019) perform text normalization on English and Russian using an RNN; the system achieves acceptable performance only when it is constrained using finite-state covering grammars. Similar findings were obtained in a Google-sponsored Kaggle competition (2017).

Recent work from Apple Inc. (Pusateri et al. 2017) describes the Siri *inverse text normalization* system, treating this as a tagging problem with a very rich tag set that might be understood as a sort of covering grammar.

Number data:
https://github.com/google-research-datasets/uninum

Homograph data:
https://github.com/google-research-datasets/WikipediaHomographData

Abbreviation data:
https://github.com/google-research-datasets/WikipediaAbbreviationData

End-to-end data:
https://www.kaggle.com/c/text-normalization-challenge-english-language
https://www.kaggle.com/c/text-normalization-challenge-russian-language